REMARKS

In the Official Action mailed on **13 November 2007**, the Examiner reviewed claims 1-18, and 28-35. Examiner rejected claims 10-18 and 32-35 under 35 U.S.C. § 101. Examiner rejected claims 1-4, 6, 7, 10-13, 15, 16, 28, 29, 31-33, and 35 under 35 U.S.C. § 102(b) based on Kwong et al. (USPN 6,289,506, hereinafter "Kwong"). Examiner rejected claims 5, 14, 30, and 34 under 35 U.S.C. § 103(a) based on Kwong. Examiner rejected claims 8 and 17 under 35 U.S.C. §103(a) based on Kwong, and Kilis (USPN 5,491,821, hereinafter "Kilis"). Examiner rejected claims 9, and 18 under 35 U.S.C. § 103(a) as being unpatentable over Kwong, and Evans et al. (USPN 5,805,899, hereinafter "Evans").

Rejection under 35 U.S.C. § 101

Claims 10-18 and 32-35 were rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. Applicant has amended the claims, changing the language from "computer-readable storage medium" to "computer-readable storage device" as per the Examiner's recommendation, and respectfully submits that these amendments obviate the rejection.

Rejections under 35 U.S.C. § 102(b)

Claims 1-4, 6-7, 10-13, 15-16, 28-29, 31-33, and 35 were rejected under 35 U.S.C. § 102(b) as being anticipated by Kwong. Applicant respectfully disagrees. Applicant respectfully points out that Kwong discloses optimizing Java performance by selecting a subset of program methods from the program code for optimization and native code compilation. This native code can be built into a

dynamic linked library. Alternately, earlier native compiled code can be de-compiled back into byte-code format. Any precompiled methods can be run concurrently with the rest of the Java byte codes of the Java application on a virtual machine (VM) (see Kwong, col. 9, lines 33-37; col. 9, lines 56-62; col. 10, lines 56-67; and col. 8, lines 38-43). This process can be repeated to further refine and optimize the program (see Kwong, col. 8, lines 46-50 and Fig. 7).

In contrast, embodiments of the present invention: (1) obtain an intermediate representation (IR) for the application program which performs calls to native code methods and callbacks into the VM (see Fig. 2, step 208, of the specification); (2) obtain an intermediate representation for selected native code methods by decompiling the native code into the IR (see Fig. 2, step 206, of the specification); (3) **combine both intermediate representations, where this combining involves inlining native code IR into call sites in the application IR** (see instant application, Fig. 2, step 210, and par. [0030]); and (4) generate native code from this combined IR while **performing optimization on this combined IR** (see Fig. 2, step 210, and paragraphs [0028]-[0032] of the instant application).

The Examiner has referred to Col. 10, lines 8-10 of Kwong as disclosing elements of the claim language regarding combining the intermediate representations, noting that "*the Java application now comprises Lib.dll 1060, A.class 1010, and B.class 1020 and may be executed on a Java VM 1080.*" Applicant respectfully points out that Lib.dll, A.class, and B.class together constitute a **juxtaposition of three elements** – *Lib.dll, A.class* and *B.class* of the code, and **they can be optimized only individually, not as a single combined entity.** It is not possible to optimize any of the individual elements **based on any other** of the elements. Applicant respectfully emphasizes that embodiments of the present invention **combine the equivalent elements into a single**

intermediate representation and optimize this combined single intermediate representation.

This is beneficial because embodiments of the present invention generate a combined IR that provides additional information to the optimization process that would not be available in the system of Kwong. Consider, for example, a case where a variable, *var*, is set within a native code method, and execution of a program segment in the application code depends upon the value of this variable, *var*. In embodiments of the present invention, it is possible to optimize the application code based on the value of *var* in the native code, since they are both combined into a single IR and available as one entity to the optimization process. This is not possible in the system of Kwong, even with optimization, because the optimization process is done separately for the native code and for the application code, and not for an integrated representation that combines both the native code as well as the application code.

There is nothing in Kwong, either explicit or implicit, that discloses generating intermediate representations for both the application program as well as the native code methods, **combining both the intermediate representations, and performing an optimization on this combined intermediate representation.** Hence, it is not possible to use the system of Kwong to improve the optimization process by using additional information from the combined IR using both the application program IR as well as the native code IR.

Accordingly, Applicant has amended claims 1, 10, 28 and 32 to clarify that combining the intermediate representations for the native code methods and the application program involves inlining the native code methods into call sites in the application. No new matter has been added.

Hence, Applicant respectfully submits that independent claims 1, 10, 28, and 32 are in condition for allowance. Applicant also submits that claims 2-9,

which depend upon claim 1, claims 11-18, which depend on claim 10, claims 29-31, which depend on claim 28, and claims 33-35, which depend on claim 32, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

CONCLUSION

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By /Anthony Jones/
Anthony Jones
Registration No. 59,521

Date: 14 January 2008

Anthony Jones
Park, Vaughan & Fleming LLP
2820 Fifth Street
Davis, CA 95618-7759
Tel: (530) 759-1666
Fax: (530) 759-1665
Email: tony@parklegal.com